SAWYER LAW GROUP LLP
2465 E. BAYSHORE ROAD SUITE 406
PALO ALTO, CA 94303
TELEPHONE: (650) 493-4540
FACSIMILE: (650) 493-4549

FACSIMILE TRANSMITTAL

Date:	January 26, 2006
To:	Examiner Anh Ly **GROUP ART UNIT: 2172**
Organization:	United States Patent and Trademark Office
Fax Number:	(571) 273-8300
Telephone Number:	(571) 272-4039
From:	Kelvin M. Vivian
Re:	USSN 09/757,427 Our Ref.: 1852P

Enclosed is the response to the Notice of Non-Compliant Appeal Brief, along with the Substitute Brief for the above-referenced case.

If you have any questions or need further information, please contact us.

This is page 1 of 20 page(s).

CONFIDENTIALITY NOTE:

The information contained in this facsimile (FAX) message is legally privileged and confidential information intended only for the use of the receiver or firm named above. If the reader of this message is not the intended receiver, you are hereby notified that any dissemination, distribution or copy of this FAX is strictly prohibited. If you have received this FAX in error, please immediately notify the sender at the telephone number provided above, and return the original message to the sender at the address above via the United States Postal Service. Thank you.

TRANSMITTAL FORM	Attorney Docket No. CA920000015/1852P

In re the application of: **Ian R. FINLAY et al.** Confirmation No: **8482**

Serial No: **09/757,427** Group Art Unit: **2172**

Filed: **January 10, 2001** Examiner: **Ly, Anh**

For: **Query Execution In Query Processing Systems**

ENCLOSURES ***(check all that apply)***

□	Amendment/Reply	□	Assignment and Recordation Cover Sheet	□	After Allowance Communication to Group
	□ After Final	□	Part B-Issue Fee Transmittal	□	Notice of Appeal
□	Information disclosure statement	□	Letter to Draftsman	■	Substitute Appeal Brief
	□ Form 1449	□	Drawings	□	Status Letter
	□ (X) Copies of References	□	Petition	□	Postcard
□	Extension of Time Request *	□	Fee Address Indication Form	□	Other Enclosure(s) *(please identify below)*:
□	Express Abandonment	□	Terminal Disclaimer		
□	Certified Copy of Priority Doc	□	Power of Attorney and Revocation of Prior Powers		
■	Response to Notification of Non-Compliant Appeal Brief	□	Change of Correspondence Address		
□	Response to Missing Parts		*Extension of Term: Pursuant to 37 CFR 1.136, Applicant petitions the Commissioner to extend the time for response for xxxxxx month(s), from to .		
	□ Executed Declaration by Inventor(s)				

CLAIMS

FOR	Claims Remaining After Amendment	Highest # of Claims Previously Paid For	Extra Claims	RATE	FEE
Total Claims	18	20	0	$ 50.00	$ 0.00
Independent Claims	6	6	0	$200.00	$ 0.00
				Total Fees	$ 0.00

METHOD OF PAYMENT

□ Check no. ______ in the amount of $ __________ is enclosed for payment of fees.

□ Charge $ ______ to Deposit Account No. _______ (Account Holder Name) for payment of fees.

■ Charge any additional fees or credit any overpayment to Deposit Account No. 09-0460 (IBM Corporation).

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Attorney Name	Kelvin M. Vivian, Reg. No. 53,727
Signature	
Date	January 26, 2006

CERTIFICATE OF TRANSMISSION/MAILING

I hereby certify that this correspondence is being facsimile transmitted to the USPTO or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Mail Stop Appeal Brief-Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on January 26, 2006.

Type or printed name	Kym Moore
Signature	

Attorney Docket: CA920000015US1/1852P

CERTIFICATE OF FACSIMILE

I hereby certify that this correspondence is being faxed to Examiner Anh Ly (571-273-8300) at the USPTO, on **January 26, 2006.**

Kym Moore

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In Re Application of:	Date: January 26, 2006
Ian R. FINLAY et al.	Confirmation No. 8482
Serial No.: 09/757,427	Group Art Unit: 2172
Filed: January 10, 2001	Examiner: Ly, Anh

For: QUERY EXECUTION IN QUERY PROCESSING SYSTEMS

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

SUBSTITUTE BRIEF ON APPEAL

In response to the Office Communication mailed December 27, 2005, Appellant provides this Substitute Brief on Appeal including "Real Party Interest and Related Appeals and Interferences," as set forth in 37 CFR 41.37 (c).

(1) Real Party in Interest

The real party in interest is International Business Machines Corporation of Armonk, New York by virtue of an assignment from the inventors recorded in the U.S. Patent Office on January 11, 2001, reel no. 011458, frame no. 0933.

(2) Related Appeals and Interferences

There are no related appeals or interferences known to the Appellant.

(3) Status of Claims

Claims 1-18 are pending in the present application, and stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,363,387 to Ponnekanti et al. ("Ponnekanti").

Claims 1-18 are being appealed.

(4) Status of Amendments

There are no unentered amendments.

(5) Summary of Claimed Subject Matter

Independent claim 1 recites a method for processing a database query on a set of data stored on a plurality of data pages in a database management system. The method includes utilizing a query processor to call a data manager and request the return of data from the set of data. Specification, page 6, lines 15-20; FIG 1. The method further includes allowing the data manager to locate query-specified data on a stabilized data page and make a determination regarding the query-specified data. Specification, page 6, lines 18-20; FIG 1. The method further includes utilizing the data manager to write the query-specified data on the stabilized data page to a buffer based on the determination while maintaining the stabilization of the data page. Specification, page 7, lines 12-20; FIG 1. The method further includes utilizing the query processor to retrieve the query-specified data from the buffer. Specification, page 8, lines 12-18; FIG 1.

Independent claim 6 recites a system for processing a database query that, in essence, implements the method of claim 1.

Independent claim 11 recites a computer readable medium containing program instructions for processing a database query that, in essence, implements the method of claim 1.

Independent claim 16 recites a method for processing a database query on a set of data stored on a plurality of data pages in a database management system. The method includes utilizing a query processor to call a data manager and request the return of data from the set of data. Specification, page 6, lines 15-20; FIG 1. The method further includes allowing the data manager to locate query-specified data by locating a data page containing query-specified data, stabilizing the data page and accessing the data page. Specification, page 7, lines 1-20; FIG 1. The method further includes utilizing the data manager to make a determination regarding the query-specified data and to write the query-specified data on the stabilized data page to a buffer based on the determination while maintaining the stabilization of the data page. Specification, page 7, lines 16-20; FIG 1. The method further includes utilizing the query processor to retrieve the query-specified data from the buffer. Specification, page 8, lines 12-18; FIG 1.

Independent claim 17 recites a system for processing a database query that, in essence, implements the method of claim 16.

Independent claim 18 recites a computer readable medium containing program instructions for processing a database query that, in essence, implements the method of claim 16.

(6) Grounds of Rejection to be Reviewed on Appeal

Appellant requests review as to claim 1-18 and their rejection under 35 U.S.C.§ 103(a) as being unpatentable over Ponnekanti.

(7) Argument

A. Summary of the Applied Rejections

In the Final Office Action, the Examiner rejected claims 1-18 under 35 U.S.C. §103(a) as being unpatentable over Ponnekanti et al. (U.S. Patent No. 6,363,387). In so doing, the Examiner stated:

> With respect to claim 1, Ponnekanti discloses utilizing a query processor to call a data manager and request the return of data from the set of data (database table storing data pages for query processing as data manager: col. 8, lines 6-20; and query processing calls to access data page table as a set of data: col. 13, lines 8-14); allowing the data manager to locate query-specified data on a stabilized data page (data page storing in the table with indexing: col. 9, lines 1-13 and col. 10, lines 28-32) and make a determination regarding the query-specified data (locating data page and maintaining the data page: col. 8, lines 38-46); utilizing the data manager to write the query-specified data on the stabilized data page to a buffer based on the determination while maintaining the stabilization of the data page (manipulation or operations on index page or data page in the table: col. 16, lines 5-30; also see buffer manager for data page: col. 10, lines 46-67); and utilizing the query processor to retrieve the query-specified data from the buffer (col. 1, lines 40-50: extracting data page from database table where is storing data page; and col. 10, lines 52-67 and col. 11, lines 1-7).
>
> Ponnekanti does not clearly disclose "to write the query-specified data on the stabilized of data page." However, Ponnekanti discloses the manipulating operations on the data page such as inserting, modifying and deleting (col. 16, lines 5-30, col. 2, lines 11-32 and col. 9, lines 45-60).
>
> Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to utilize the manipulation operations for data page storing in the table (col. 16, lines 5-30) for reducing overhead as taught by Ponnekanti (col. 3, lines 22-30) because it would have made method for processing data page in the table being

> optimizing the query processing (col. 4, lines 35-44) and reducing the overhead of locking and increasing the concurrency, a particular performance advantage of the technique in querying data page (col. 10, lines 21-26) and this method would provide the index manager to locate the index and continue scanning the table at the next record row (col. 16, lines 20-40) in the fetching index-data environment.

B. The Cited Prior Art - Ponnekanti

Ponnekanti is directed to a method for minimizing locking to optimize concurrency. In Ponnekanti, "locking is done after reading the data row, taking into account the status bits of data row and whether the data row qualifies." Col. 4, lines 36-43. During an index scan, Ponnekanti discloses deferred locking where the Index Manager returns RIDs to the Data Layer without acquiring locks on them during scans that have scan arguments on columns not in the index. The Data Layer subsequently qualifies the data row and determines whether locking is really necessary. Thus, the locking is done after the Data Layer reads the data row and determines that the data row qualifies. By determining whether an index or data row qualifies before applying the lock on the row, Ponnekanti reduces locking overhead and increases concurrency.

C. Independent Claims 1, 6, 11, 16, 17 and 18 Are Allowable Over the Cited Reference.

Claim 1 recites a method for processing a database query on a set of data stored on a plurality of data pages in a database management system. The method includes utilizing a query processor to call a data manager and request the return of data from the set of data, and allowing the data manager to locate query-specified data on a stabilized data page and make a determination regarding the query-specified data. The method further includes utilizing the data manager to write the query-specified data on the stabilized data page to a buffer based on the determination while maintaining the stabilization of the data page, and utilizing the query processor to retrieve the query-specified data from the buffer.

Appellant respectfully submits that Ponnekanti fails to teach or suggest the present invention, as recited in each of claims 1, 6, 11, 16, 17 and 18. In particular, Ponnekanti does not disclose or teach "utilizing the data manager to write the query-specified data on the stabilized data page to a buffer . . . while maintaining the stabilization of the data page" and "utilizing the query processor to retrieve the query-specified data from the buffer," as recited in each of claims 1, 6, 11, 16-18.

In the present invention as claimed, during a table scan, the data manager stabilizes, i.e., latches, a data page, locates query-specified data on the stabilized data page, writes the query-specified data from the stabilized data page to a buffer (instead of returning it directly to the query processor) while maintaining the latch on the data page, and continues reading the rows on the data page. The query processor then fetches the data from the buffer. By writing the query-specified data to a buffer, query processing is more efficient because the number of page stabilizations is reduced.

Ponnekanti does not teach or suggest optimizing query processing *in such a manner*. In Ponnekanti, "the action taken by a scan depends on whether the row qualifies and the status of the row status bits. Qualification is done while holding a latch." (Col. 13, lines 15-20). Therefore, qualification entails checking the row status bits and the data. If the row status bits are not set or if the row status bit is "update," and if the data qualifies, the lock manager grants a LOCK on the row and the data manager returns the data row. (Col. 13, lines 23-27; 52-56). Otherwise, the row is either skipped or a LOCK_INSTANT request is processed. (Col. 13, line 28 col. 14, line 2).

Thus, in Ponnekanti, query-specified data on a latched page, i.e., is locked and *returned* to the query processor if the data qualifies and if the row status bit is "unset" or "update." Ponnekanti makes no mention or suggestion of *writing* "the query-specified data to a buffer" after such data has been located and qualified on a stabilized page, nor does Ponnekanti teach or suggest "utilizing the query processor to retrieve the query-specified data from the buffer," as recited in claims 1, 6, 11, and 16-18 (emphasis added).

In the Final Office Action, the Examiner cites various portions of Ponnekanti which purportedly support his position that Ponnekanti teaches or suggests these features. Appellant respectfully disagrees. The cited portions are listed below with a summary of their content.

Column, lines cited **(in the order presented in the Final Office Action and in the Advisory Action)**	**Summary**
Column 16, lines 5-30	Describes the process for race conditions due to deferred locking. Where an index manager returns the row ID of a qualifying row to a data layer, the data layer must check with the index manager after the data layer latches the data page to determine whether any changes to the index page have occurred. Lines 5-30 describe what the data layer and index manager do in this situation. It does not teach or suggest writing query-specified data to a buffer on a stabilized page.
Column 10, lines 46-47 [sic] --- more likely lines 66-67	Describes generally that latches are held on "kept" pages, which are pages "guaranteed by the database system's buffer manager to not be replaced from

	the buffer cache." The fact that a buffer cache exists or that kept pages are in the buffer cache does not teach or suggest writing query-specified data to a buffer on a stabilized page.
Column 1, lines 40-50	Discusses database management systems in general. Does not mention or suggest the query processor *retrieving* query-specified data from the buffer.
Column 10, line 52 to Column 11, line 7	Describes latching mechanisms in general. Does not mention or suggest the query processor *retrieving* query-specified data from the buffer.
Column 2, lines 11-32	Discusses database management systems in general and in particular, locking schemes.
Column 9, lines 45-60	Describes generally how a node overflow (in a B-Tree node) is handled by splitting the node.

Appellant respectfully submits that none of those portions (nor any other portions of Ponnekanti) discloses or teaches writing "the query-specified data on the stabilized data page to a buffer . . . while maintaining the stabilization of the data page" and retrieving "utilizing the query processor to retrieve the query-specified data from the buffer," as recited in each of claims 1, 6, 11, and 16-18.

In the Advisory Action, the Examiner asserts a somewhat different argument stating that the Access Methods in Figure 2A, item 269 is analogous to the present invention's query processor and that the Lock Manager is analogous to the present invention's data manager. "The Access Methods (as query manager or query processor) calls the Lock Manager (as data manager) to acquire and release locks and latches. Thus, Lock Manager is updating or writing the data page in the buffer based on the query-specified, which is carried out by Access Methods, which is fetching the data page stored in the buffer (col. 10, lines 48-67)."

Appellant respectfully submits that the Examiner's reasoning is unsupported by Ponnekanti. Ponnekanti clearly states that the Access Methods are "*lower-level routines* . . . for carrying out the query-specified operation, such as fetching relevant information (e.g., row 255) from the database table 250." (Col. 7, lines 46-49). Moreover, Ponnekanti explicitly states:

> The Lock Manager is employed for providing interfaces for acquiring and releasing conditional and unconditional locks. . . . The Access Methods (module) calls the Lock Manager/Latch Manager to acquire and release locks and latches. Access Methods, as previously described, provides interfaces functions, such as performing a single table scan or an index scan and inserting a single row, deleting a single row specified by RID, and updating a row specified by RID given its new image, and the like. Access Methods employ an "Index Manager" which manages indices and a "Data Layer" (module) which manages data pages. A "Query Processing Layer" sits on top of Access Methods. It is responsible for splitting complex SQL statements into simpler operations that can be handled by the Access Methods. Col. 11, line 59 to col. 12, line 8.

Thus, according to Ponnekanti, the Access Methods, as opposed to the Lock Manager, is more akin to the data manager of the present invention in that: (1) it is called by the Execution Unit 268 to fetch query-specified data from a database table, (2) it requests latches for pages, (3) it locates query-specified data on latched pages, (4) requests locks for the rows containing query-specified data, and (5) it returns query-specified data to the Execution Unit 268. Nothing in Ponnekanti teaches or suggests that any of these functions are *performed* by the Lock Manager. Accordingly, Appellant respectfully submits that Ponnekanti's Lock Manager cannot teach or suggest the data manager of the present invention.

Moreover, even if the Lock Manager were to be considered the data manager (which Appellant does not concede), there is no teaching or suggestion in Ponnekanti that the Lock Manager writes "the query-specified data on the stabilized page to a buffer" and that the Access Methods then retrieves "the query-specified data from the buffer." The portion of Ponnekanti purportedly supporting this assertion only describes locking and latching conceptually. (Col. 10, lines 48-67). It makes no mention of the Lock Manager or of the Access Methods.

Appellant respectfully submits that Ponnekanti fails to disclose or teach the combination of elements recited in 1, 6, 11 and 16-18. Accordingly, claims 1, 6, 11 and 16-18 should be in condition for allowance. Claims 2-5, 7-10 and 12-15 depend on claims 1, 6 and 11, respectively and, therefore, incorporates the limitations of those claims. Appellant respectfully submits that each of claims 2-5, 7-10 and 12-15 are improperly rejected for reasons similar to those discussed above. Therefore, Appellant respectfully submits that claims 2-5, 7-10 and 12-15 are also allowable.

D. The Examiner has failed to establish a *prima facie* case of obviousness

To establish a *prima facie* case of obviousness, the Examiner must make three basic showings. First, there must be some suggestion or motivation, either in the references or in the prior knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, and not based on Applicant's disclosure. *In re Vaeck*, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991).

The Examiner has failed to make any of the three basic showings and consequently, no *prima facie* case of obviousness has been established. Thus claims 1-18 are improperly rejected under §103(a).

E. Summary of Arguments

Appellant respectfully submits that Ponnekanti does not disclose or teach "utilizing the data manager to write the query-specified data on the stabilized data page to a buffer . . . while maintaining the stabilization of the data page" and "utilizing the query processor to retrieve the query-specified data from the buffer," as recited in each of claims 1, 6, 11, 16-18. Accordingly, for the reasons set forth above, Appellant respectfully submits that claims 1-18 are allowable over Ponnekanti. Appellant respectfully requests that the final rejection of claims 1-18 be reversed.

Please charge any fee that may be necessary for the continued pendency of this application to Deposit Account No. 09-0460 (IBM Corporation).

Respectfully submitted,
SAWYER LAW GROUP LLP

January 26, 2006
Date

Kelvin M. Vivian
Attorney for Applicant
Reg. No. 53,727
(650) 493-4540

Appendix of Claims

1. (Previously Presented) A method for processing a database query on a set of data stored on a plurality of data pages in a database management system, the method comprising the steps of:

a) utilizing a query processor to call a data manager and request the return of data from the set of data;

b) allowing the data manager to locate query-specified data on a stabilized data page and make a determination regarding the query-specified data;

c) utilizing the data manager to write the query-specified data on the stabilized data page to a buffer based on the determination while maintaining the stabilization of the data page; and

d) utilizing the query processor to retrieve the query-specified data from the buffer.

2. (Original) The method of claim 1 wherein the determination involves determining whether the query-specified data is to be ignored, consumed, or returned to the query processor.

3. (Original) The method of claim 2 wherein the determination is that the query-specified data is to be returned to the query processor.

4. (Original) The method of claim 3 wherein the set of data is stored on a plurality of pages and step b) further comprises:

b1) locating a page containing query-specified data;

b2) stabilizing the page; and

b3) accessing the page.

5. (Original) The method of claim 4 wherein step c) further comprises:

c1) maintaining the stabilization of the page, while the data manager writes all the query-specified data on the page to the buffer.

6. (Previously Presented) A system for processing a database query on a set of data stored on a plurality of data pages in a database management system, the system comprising:

a data manager;

a query processor comprising means to call the data manager and request the return of data from the set of data;

means for allowing the data manager to locate query-specified data on a stabilized data page and make a determination regarding the query-specified data;

means for utilizing the data manager to write the query-specified data on the stabilized data page to a buffer based on the determination while maintaining the stabilization of the data page; and

means for utilizing the query processor to retrieve the query-specified data from the buffer.

7. (Original) The system of claim 6 wherein the determination involves determining whether the query-specified data is to be ignored, consumed, or returned to the query processor.

8. (Original) The system of claim 7 wherein the determination is that the query-specified data is to be returned to the query processor.

9. (Original) The system of claim 8 wherein the set of data is stored on a plurality of pages and the means for allowing the data manager to locate query-specified data further comprises:

means for locating a page containing query-specified data;

means for stabilizing the page; and

means for accessing the page.

10. (Original) The system of claim 9 wherein the means for utilizing the data manager to write the query-specified data to a buffer further comprises:

means for maintaining the stabilization of the page, while the data manager writes all the query-specified data on the page to the buffer.

11. (Previously Presented) A computer readable medium containing program instructions for processing a database query on a set of data stored on a plurality of data

pages in a database management system, the program instructions comprising the steps of:

a) utilizing a query processor to call a data manager and request the return of data from the set of data;

b) allowing the data manager to locate query-specified data on a stabilized data page and make a determination regarding the query-specified data;

c) utilizing the data manager to write the query-specified data on the stabilized data page to a buffer based on the determination while maintaining the stabilization of the data page; and

d) utilizing the query processor to retrieve the query-specified data from the buffer.

12. (Original) The computer readable medium of claim 11 wherein the determination involves determining whether the query-specified data is to be ignored, consumed, or returned to the query processor.

13. (Original) The computer readable medium of claim 12 wherein the determination is that the query-specified data is to be returned to the query processor.

14. (Original) The computer readable medium of claim 13 wherein the set of data is stored on a plurality of pages and step b) further comprises:

b1) locating a page containing query-specified data;

b2) stabilizing the page; and

b3) accessing the page.

15. (Original) The computer readable medium of claim 14 wherein step c) further comprises:

c1) maintaining the stabilization of the page, while the data manager writes all the query-specified data on the page to the buffer.

16. (Previously Presented) A method for processing a database query on a set of data stored on a plurality of data pages in a database management system, the method comprising the steps of:

a) utilizing a query processor to call a data manager and request the return of data from the set of data;

b) allowing the data manager to locate query-specified data by:

b1) locating a data page containing query-specified data;

b2) stabilizing the data page; and

b3) accessing the data page;

c) utilizing the data manager to make a determination regarding the query-specified data and to write the query-specified data on the stabilized data page to a buffer based on the determination while maintaining the stabilization of the data page; and

d) utilizing the query processor to retrieve the query-specified data from the buffer.

17. (Previously Presented) A system for processing a database query on a set of data stored on a plurality of data pages in a database management system, the system comprising:

a data manager;

a query processor comprising means to call the data manager and request the return of data from the set of data;

means for allowing the data manager to locate query-specified data comprising means for locating a data page containing query-specified data, means for stabilizing the data page, and

means for accessing the data page;

means for utilizing the data manager to make a determination regarding the query-specified data and to write the query-specified data on the stabilized data page to a buffer based on the determination while maintaining the stabilization of the data page; and

means for utilizing the query processor to retrieve the query-specified data from the buffer.

18. (Previously Presented) A computer readable medium containing program instructions for processing a database query on a set of data stored on a plurality of data pages in a database management system, the program instructions comprising the steps of:

a) utilizing a query processor to call a data manager and request the return of data from the set of data;

b) allowing the data manager to locate query-specified data by:

b1) locating a data page containing query-specified data;

b2) stabilizing the data page; and

b3) accessing the data page;

c) utilizing the data manager to make a determination regarding the query-specified data and to write the query-specified data on the stabilized data page to a buffer based on the determination while maintaining the stabilization of the data page; and

d) utilizing the query processor to retrieve the query-specified data from the buffer.